

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ВНУТРІШНІХ СПРАВ  
КРЕМЕНЧУЦЬКИЙ ЛЬОТНИЙ КОЛЕДЖ**

**Циклова комісія технічного обслуговування авіаційної техніки**

**ТЕКСТ ЛЕКЦІЇ**

навчальної дисципліни

«Інформаційні системи контролю та діагностики газотурбінних двигунів»  
вибіркових компонент  
освітньо-професійної програми першого (бакалаврського) рівня вищої освіти

**272 Авіаційний транспорт**

**Технічне обслуговування та ремонт повітряних суден і авіадвигунів**

**за темою №5. – Подання заяв в інтелектуальних системах**

**Кременчук 2023**

**ЗАТВЕРДЖЕНО**

Науково-методичною радою  
Харківського національного  
університету внутрішніх справ  
Протокол від 30.08.2023 № 7

**СХВАЛЕНО**

Методичною радою Кременчуцького  
льотного коледжу Харківського  
національного університету внутрішніх  
справ  
Протокол від 28.08.2023 № 1

**ПОГОДЖЕНО**

Секцією Науково-методичної ради  
ХНУВС з технічних дисциплін  
Протокол від 29.08.2023 № 7

Розглянуто на засіданні циклової комісії технічного обслуговування авіаційної  
техніки, протокол від 28.08.2023 № 1

**Розробник:**

*1. Викладач циклової комісії технічного обслуговування авіаційної техніки,  
викладач-спеціаліст Самохліб Олександр Олександрович*

**Рецензенти:**

*1. Завідувач кафедри технологій аеропортів Національного авіаційного  
університету, д.т.н., професор Тамаргазін О.А.  
Викладач циклової комісії аеронавігації КЛК ХНУВС, к.т.н., с.н.с. Тягній В.Г.*

### **План лекції**

1. Підходи до подання знань.
2. Вербально-дедуктивне визначення знань. Експертні системи.
3. Дані та знання.
4. Зв'язки між інформаційними одиницями.
5. Проблема винятків.
6. Властивості та моделі знань.
7. Неоднорідність знань.
8. Області і рівні знань.
9. База знань як об'єднання простіших одиниць.
10. Бінарні предикати і тріада «об'єкт—атрибут—значення».

### **Рекомендована література:**

#### **Основна**

1. Нечипоренко О. М. Основи надійності літальних апаратів : навчальний посібник. Київ : НТУУ «КПІ», 2010. 240 с.

#### **Допоміжна**

2. Нестеренко О. В., Савенков О. І., Фаловський О. О. Інтелектуальні системи підтримки прийняття рішень : навчальний посібник. Київ : Національна академія управління, 2016. 188 с.
3. Вахнюк С.В. Технологія створення програмних та інтелектуальних систем: навчальний посібник. Суми : УАБС НБУ, 2011. 254 с.
4. Шаров С. В., Лубко Д. В., Осадчий В. В. Інтелектуальні інформаційні системи : навчальний посібник. Мелітополь: Вид-во МДПУ ім. Б. Хмельницького, 2015. 144 с.

#### **Інформаційні ресурси в інтернеті**

1. <https://openarchive.nure.ua/server/api/core/bitstreams/ced9dde2-bace-4fe0-8e5f-8a18fd4da046/content>

## ТЕКСТ ЛЕКЦІЇ

### 5.1. Підходи до подання знань

У філософії знання визначаються як "відображення об'єктивних властивостей та зв'язків світу".

*Знання є інформаційною основою інтелектуальних систем, оскільки саме вони завжди зіставляють зовнішню ситуацію зі своїми знаннями і керуються ними при прийнятті рішень. Не менш важливим є те, що знання — це систематизована інформація, яка може певним чином поповнюватися і на основі якої можна отримувати нову інформацію, тобто нові знання.*

Існує багато підходів до визначення поняття "знання". На сучасному етапі домінуючою парадигмою, що лежить в основі найвідоміших моделей подання знань у системах штучного інтелекту, можна вважати парадигму (певну сукупність ключових принципів), характерну для **символьного підходу**. Цю парадигму можна охарактеризувати як **вербально-дедуктивну**, або **словесно-логічну**, через певні чинники:

- будь-яка інформаційна одиниця задається вербально, тобто у формі, наближеній до словесної, у вигляді набору явно сформульованих тверджень або фактів;

- основним механізмом отримання нової інформації на базі існуючої є **дедукція**, тобто висновок від загального до часткового. Такий дедуктивний підхід є бездоганим з логічного погляду. В його основі лежать транзитивність імплікації (якщо з  $a$  випливає  $b$ , з  $b$  випливає  $c$ , то з  $a$  випливає  $c$ ) і дія квантора загальності (якщо деяка властивість  $P$  виконується для будь-якого елемента множини  $M$ , а

$x \in M$ , то  $P$  виконується для  $x$ ).

Але вербально дедуктивне задання знань не є повним, оскільки:

- дедуктивний висновок не виступає єдиною можливістю. Мислення людини багато в чому є рефлексивним, інтуїтивним. Воно, як правило, спирається на підсвідомі процеси. Людина часто робить висновки за аналогією, асоціацією. Ці висновки не завжди вірні, але вони істотно доповнюють процеси дедуктивного мислення. Без підсвідомого мислення стає неможливим (будь-яке відкриття — як правило, підсвідоме породження гіпотези, яка потім перевіряється дедуктивним або експериментальним шляхом);

- далеко не всі знання є вербальними. Так, жодне твердження не зберігається в пам'яті людини явно. Відомо, що в основі діяльності мозку людини лежить передача сигналів між нервовими клітинами. Часто людина не може сформулювати свої знання. Наприклад, будь-хто знає, що таке "стіл". Але якщо попросити людину дати визначення цього поняття, у неї можуть виникнути проблеми. Таким чином, поняттями можна оперувати і не знаючи чіткого їх визначення. Типовими є слова: "Я не можу пояснити чому, але мені здається".

Тому необхідно розвивати інші моделі знань, окрім вербально-дедуктивних. Наприклад, у рамках **конекціоністського підходу** можна розглядати моделі на основі однорідного поля знань. **Однорідним полем знань**

називається сукупність простих однорідних елементів, які обмінюються між собою інформацією: нові знання народжуються на основі певних процедур, визначених над полем знань.

## 5.2. Вербально-дедуктивне визначення знань

У рамках вербально-дедуктивної парадигми можна навести таке визначення знань.

*Знаннями інтелектуальної системи називається трійка  $\langle F, R, P \rangle$ , де  $F$  — сукупність явних фактів, які зберігаються в пам'яті системи в явному вигляді,  $R$  — сукупність правил виведення, які дозволяють на основі відомих знань набувати нових знань,  $P$  — сукупність процедур, які визначають, яким чином слід застосовувати правила виведення.*

*Базою знань (БЗ) інтелектуальної системи називатимемо сукупність усіх знань, що зберігаються в пам'яті системи.*

У базах знань прийнято розрізняти екстенціональну та інтенціональну частини.

Екстенціональною частиною бази знань називається сукупність усіх явних фактів, інтенціональною частиною — сукупність усіх правил виведення та процедур, за допомогою яких з існуючих фактів можна виводити нові твердження.

Приклад. Розглянемо таку базу знань.

**Ф1.** Заєць їсть траву. **Ф2.** Вовк їсть м'ясо. **Ф3.** Заєць є м'ясом.

**П1.** Якщо  $A$  є м'ясом, а  $B$  їсть м'ясо, то  $B$  їсть  $A$ .

**П2.** Якщо  $A$  їсть м'ясо, то  $A$  є хижаком.

**П3.** Якщо  $A$  їсть  $B$ , то  $B$  є жертвою.

**П4.** Якщо хижак вмирає, жертва швидко розмножується.

**П5.** Якщо жертва вмирає, хижак вмирає.

Даний приклад є фрагментом неформалізованого опису моделі Лотки-Вольтерра, добре відомої в екології. Літерами "Ф" позначені явні факти, які відображають елементарні знання. Їх безпосередній аналіз дозволяє експертній системі відповідати на найпростіші запитання, такі, як "Вовк їсть м'ясо?". Для позитивної відповіді досить просто знайти відповідний факт в екстенціональній частині бази знань.

Принципово іншим є запитання типу "Вовк є хижаком?". Відповідь на нього на основі простого пошуку знайти неможливо: такого явного факту в екстенціональній частині бази знань немає. Тому для відповіді на це запитання необхідно застосовувати правила виведення, які дозволяють на основі існуючих явних фактів набувати нових знань, тобто нових фактів. Такі правила позначені у нашому прикладі літерою "П". Наприклад, для відповіді на поставлене запитання досить знання факту **Ф2** і правила **П2**.

Безумовно, до бази знань можна було б відразу включити твердження "Вовк є хижаком". Але легко побачити: якщо, крім вовків, розглядати інших хижаків (ведмедів, лисиць та ін.), то безпосередньо включати до бази знань явні факти, що вони є хижаками, недоцільно. Все, що може бути виведене логічним шляхом, як правило, не варто оголошувати фактами. Надалі називатимемо явні

факти просто фактами, якщо це не викликає непорозуміння.

Нарешті, **процедури** визначають, яким саме чином слід застосовувати правила. Часто ці процедури є складовою мови, якою програмується експертна система (це стосується насамперед спеціалізованих мов ШП, таких як Лісп, Пролог, Пленер).

### 5.3.

### Експертні системи

**База** знань, наведена в п. 3.2, може лягти в основу *експертної системи*, тобто інформаційної системи, яка здійснює дедуктивне виведення на основі наявних знань. Існують різні визначення експертних систем, основна суть яких зводиться до такого. **Експертні системи** — це інтелектуальні програмні засоби, здатні у діалозі з людиною одержувати, накопичувати та коригувати знання із заданої предметної галузі, виводити нові знання, розв'язувати на основі цих знань практичні задачі та

*пояснювати хід їх розв'язку.*

Експертні системи акумулюють знання експертів — провідних спеціалістів у даній предметній галузі. В основі роботи експертних систем лежить дедуктивне виведення нових тверджень з існуючих. Типове застосування експертних систем — консультації для фахівців середньої кваліфікації і неспеціалістів у тій галузі, для якої вона розроблена.

Створити універсальну експертну систему неможливо. По-перше, це пов'язано з "прокляттям розмірності". По-друге, експертна система повинна акумулювати знання людей-експертів, а "універсальних" експертів не існує і не може існувати. По-третє, жодне логічне виведення не може замінити інтуїцію та досвід експерта.

Можна лише експертні системи, які належать до конкретної предметної галузі. Остання передбачає лімітований набір явищ і понять певної сфери людської діяльності та обмежене коло задач, які вирішуються у цій галузі. Існує чимало експертних систем у таких сферах як медична і технічна діагностика, пошук корисних копалин, юриспруденція, аналіз інвестицій і комерційних ризиків і т. п. У створенні експертних систем повинні брати участь фахівці як мінімум двох категорій:

- **експерт**, що є висококваліфікованим фахівцем у даній предметній області, знання якого потрібно передати експертній системі;

- **інженер знань**, завдання якого — формалізувати знання експерта і привести їх до вигляду, придатного для занесення до бази знань.

Серйозна проблема у даному випадку пов'язана з тим, що знання експертів важко формалізувати. Експерт часто не може сформулювати свої знання у явному вигляді, робить правильні висновки, але не може пояснити як саме він їх робить. Якщо ж експерт і формулює певні правила виведення, вони далеко не завжди відзначаються точністю та адекватністю. З цього випливає ряд інших труднощів, які так чи інакше виявляють себе на етапі формалізації знань або застосування готових експертних систем.

### 5.4.

### Дані та знання

Відомо, що сучасний етап розвитку інформатики характеризується

еволюцією моделей даних у напрямі переходу від традиційних (реляційна, ієрархічна, мережева) до моделей знань. Знання, як будь-яку інформацію, можна вважати даними. Але знання є високоорганізованими даними, для яких характерна певна внутрішня структура та розвинуті зв'язки між різними інформаційними одиницями. Іншим принциповим аспектом є те, що інформаційні системи, які ґрунтуються на знаннях, повинні мати можливість отримувати нові знання на основі існуючих. У цьому ми вже пересвідчилися на прикладах дедуктивного виведення нових знань з явних фактів.

**Екстенціональна частина** БЗ складається з фактів, які запам'ятовуються явно, інтенціональну — з правил, які дозволяють отримувати нові факти. Наявність розвинутої інтенціональної частини є однією з основних рис, які відрізняють знання від традиційних даних. **Інтенціональним відношенням**, яке описує базу даних, часто називають правило або сукупність правил, яким підпорядковується кожний запис у базі даних.

Приклад. Розглянемо базу даних (БД) деканату, що містить дані про те, які курси прослухав кожний студент. Відомо, що жоден студент не має права слухати курс "Бази даних", якщо він не прослухав курсу "Основи програмування". Нехай у базі даних зберігається інформація про Іванова, Петрова та Сидорова, які прослухали обидва курси.

*Екстенціональна частина:*

Прізвище	Дисципліна
Іванов	Бази даних
Петров	Бази даних
Сидоров	Бази даних

Інтенціональна БД (або її вже можна назвати БЗ) може мати такий вигляд:

Правило: Якщо Прослухав(X, Бази даних), то Прослухав(X, Основи програмування).

Наявність такого правила дозволяє скоротити базу знань: твердження, істинність яких можна встановити за допомогою правил, не обов'язково запам'ятовувати в явному вигляді.

Пошук в інтенціональній БЗ складається з двох етапів:

- пошук потрібного факту в екстенціональній частині;
- дедуктивне виведення факту на основі правил інтенціональної частини.

Реалізовувати інтенціональні правила можна навіть у рамках реляційної моделі даних шляхом програмування відповідних запитів.

Знання можуть бути **неповними**. Це означає, що для доведення або спростування певного твердження може не вистачати інформації. У багатьох системах логічного виведення прийнято **постулат замкненості світу**: на запит про істинність деякого твердження система відповідає "так" тоді і тільки тоді, коли його можна довести; якщо ж довести неможливо, система відповідає "ні". Водночас "неможливо довести через нестачу інформації" і "доведено, що ні" — це зовсім не те саме. З огляду на це бажано, щоб експертна система запитувала у користувача профакти, яких не вистачає.

Знання можуть бути **недостовірними**. Наприклад, на виготовлення

продукції можуть впливати випадкові чинники (об'єктивна невизначеність) або експерт може бути не зовсім упевненим у деякому факті чи правилі (суб'єктивна невизначеність). Ненадійність знань і недостовірність наявних фактів обов'язково повинні враховуватися в процесі логічних побудов. Звичайно, можна було б просто відкидати факти та правила виведення, які викликають сумнів, але довелося би відмовитися від цінної інформації. Крім того, в експертних системах часто доводиться мати справу з

неточно визначеними поняттями, такими, як "великий", "маленький" тощо.

### 5.5. Зв'язки між інформаційними одиницями

У базі знань можуть зберігатися відомості про різні об'єкти, поняття, процеси і т. п., тобто відповідні описи, які називаються **інформаційними одиницями**.

*Опис; який утворює інформаційну одиницю, розглядається як деяка абстракція реальної сутності, що існує в дійсному світі.*

Це означає що дана абстракція описує певні риси реальної сутності. Багато, щоб абстракція описувала найважливіші з них.

У реальному світі все тісно взаємопов'язане. Відповідні зв'язки повинні знайти адекватне відображення в БЗ. Існує велика група зв'язків, за допомогою яких можна описувати просторові відношення ("знизу", "зверху", "поруч", "між" і т. п.); часові відношення ("раніше", "пізніше", "під час", "одночасно" і т. п.), причинно-наслідкові відношення тощо. Ці відношення є спільними для всіх предметних областей. Логічні системи, що ґрунтуються на цих відношеннях, називаються **псевдофізичними логіками**. Використання псевдофізичних логік (так само, як і більш загальних предикатів, таких, як узагальнення та агрегація) дає можливість інтелектуальній системі поповнювати первинні описи.

Так, якщо інтелектуальна система сприймає текст: "У 1985 році почалася перебудова. У 1991 році була проголошена незалежність України", врахування часових властивостей, що визначається часовою логікою, дає змогу відповідати на запитання типу: "Що було раніше: початок перебудови чи проголошення незалежності України?"

Основними зв'язками між інформаційними одиницями є **узагальнення** та

#### **агрегація.**

**Агрегація** дозволяє задавати будову об'єкта та його складових. Наприклад, агрегація дає змогу визначити в базі знань аудиторію як об'єкт, що має вікна, двері і т. п. Вікна, в свою чергу, теж можна розглядати як агрегований об'єкт: вони мають ручки, рами і т. п. Інакше кажучи, відношення "Має", яке описує агрегацію, означає, що певний об'єкт містить у своєму складі деякий інший об'єкт. За цим відношенням закріпилася спеціальна назва **Has\_Part**. По суті, це відношення "ціле — частина".

Таким саме чином ввести і зворотне відношення ("Є частиною", або **Is\_Part**). **Узагальнення** (відношення "Є") задає ієрархію класів (екземпляр — клас —

**надклас**). Так, можна говорити, що об'єкт Вася Петров належить до класу



Студент. Відповідно інформаційна одиниця, яка описує Василя Петрова, може бути описана на основі більш загальної інформаційної одиниці Студент (точніше, вона задається як екземпляр інформаційної одиниці Студент). Клас Студент, у свою чергу, є *підкласом* типу Людина і т. п. Надклас об'єднує атрибути, або ознаки, спільні для його підкласів; підкласи можуть успадковувати ті чи інші властивості надкласів.

Із відношенням "Є" пов'язаний один з основних відомих механізмів логічного виведення — механізм **виведення за успадкуванням** (інша назва — **виведення за наслідуванням**). Суть цього механізму можна сформулювати так: якщо деяка умова виконується для всього класу, то вона виконується і для кожного представника цього класу, а також для всіх підкласів цього класу (якщо інше не задано явним чином). Інакше кажучи, екземпляри успадковують властивості класів, підкласи успадковують властивості надкласів.

В усіх системах керування базами знань повинна бути забезпечена належна підтримка успадкування. Розглянемо *приклад* [8]. У базі знань міститься така інформація:

*Усі птахи літають. Ластівка є птахом. Юкко є ластівкою.*

Маємо загальний клас "Птахи", його підклас "Ластівки" та об'єкт "Юкко", який є екземпляром класу "Ластівка". На підставі цих знань будь-яка система, що підтримує успадкування, повинна зробити висновки, що *Юкко є птахом; всі ластівки літають; Юкко теж літає.*

Насправді, замість єдиного відношення "Є" необхідно розглядати цілу систему споріднених, але не ідентичних відношень. Інакше легко припуститися логічних помилок, подібних до таких:

*Юкко є ластівкою.*

*Ластівка є видом, який вивчається натуралістами.*

*Отже, Юкко є видом, який вивчається натуралістами.*

Слід розглянути такі відношення:

- **"клас — підклас"**, яке є відношенням часткового порядку (звичайно строгого). Для цього відношення виконується властивість транзитивності;

- **"екземпляр — клас"**. Екземпляри успадковують властивості своїх класів, але саме відношення "екземпляр — клас" не є транзитивним.

"Ластівка" є екземпляром класу "Види, які вивчаються натуралістами", а "Юкко" є екземпляром класу "Ластівка", але "Юкко" у жодному разі не є екземпляром класу "Види, які вивчаються натуралістами" ("Юкко" взагалі не є видом).

Також необхідно чітко розрізняти поняття **"клас"** і **"множина"**, **"належність до класу"** та **"належність до множини"**. Клас об'єднує однотипні елементи, множина — необов'язково. Відношення "елемент — множина" також не є транзитивним. Складні системи можуть бути описані у вигляді певної канонічної форми, яка являє собою композицію двох ієрархій — **ієрархії класів** та **ієрархії об'єктів**.

**Ієрархія класів** задається відношенням *узагальнення*, **ієрархія об'єктів** —

*відношенням агрегації.*

Дійсно, з одного боку, такі об'єкти, як "ручки", "рами" та інші, утворюють новий об'єкт — "вікно". "Вікна", "двері" тощо — це об'єкти, які утворюють ще один новий об'єкт: "аудиторію". Таким чином, можна розглядати певну ієрархію об'єктів. З іншого ж боку, всі ці об'єкти є екземплярами своїх класів. Так, конкретне вікно належить до класу "Вікна", клас "Вікна" є підкласом класу "Дерев'яні вироби" і т. п. Відношення "клас — підклас" є характерним для ієрархії класів, відношення "елемент — множина" та "підмножина — множина" — до ієрархії об'єктів, а

відношення "екземпляр — клас" об'єднує обидві ієрархії.

## 5.6. Проблема винятків

З успадкуванням пов'язана дуже серйозна проблема — **проблема винятків**. Вона полягає в тому, що деякі підкласи можуть не успадковувати ті чи інші властивості надкласів. Інакше кажучи, характерні риси класу успадковуються всіма його підкласами, **крім** деяких.

Нехай відомо, що *літають всі птахи, крім пінгвінів* (існують деякі інші види птахів, які не літають, але для наших цілей це не має суттєвого значення). Якби це

твердження відразу потрапило до БЗ саме в такому вигляді, особливих проблем не виникало б (хоча і в цьому випадку слід було б передбачити належну обробку винятків).

Але, як було зазначено раніше, експерт не завжди може сформулювати свої знання в явному вигляді. Зокрема, він може не знати або не пам'ятати всіх винятків. Тому він може спочатку включити до БЗ твердження про те, що *всі птахи літають*, а потім пригадати, що *пінгвіни не літають*, і додати це до БЗ. У результаті ми могли б отримати БЗ, подібну до такої:

*Усі птахи літають. Ластівка є птахом. Юкко є ластівкою. Пінгвін є птахом.*

*Пінгвіни не літають. Бакс є пінгвіном.*

Якби три останні твердження **не були** включені до бази знань, системі просто дійшла б хибного висновку, що *Бакс літає*. Але включення даних відомостей до бази знань ще більше ускладнює ситуацію. **Система знань стає суперечливою**: з одного боку, система повинна дійти висновку, що Бакс літає, а з іншого — що Бакс не літає. У даному разі кажуть про **втрату монотонності** дедуктивної системи [8].

**Система дедуктивного виведення називається монотонною**, якщо для неї виконується така властивість: якщо з набору тверджень  $(q_1, \dots, q_n)$  випливає твердження  $V$ , то  $V$  випливає і з набору тверджень  $(q_1, \dots, q_n, r)$ .

Інакше кажучи, в монотонній системі додавання нових фактів і правил не впливає на істинність висновків, які могли бути отримані без них.

Для вирішення проблеми винятків часто використовують наступне правило:

*у разі виникнення суперечностей підклас успадковує відповідну властивість лише від найближчого попередника, тобто найближчого до нього в ієрархії*

класів.

### 5.6. Властивості та моделі знань

У [9] сформульовані такі особливості знань, які відрізняють їх від звичайних даних:

- 1) **внутрішня інтерпретованість:** кожна інформаційна одиниця повинна мати унікальне ім'я, за яким інформаційна система її знаходить, а також відповідає на запити, в яких це ім'я згадується;
- 2) **структурованість:** знання повинні мати гнучку структуру; одні інформаційні одиниці можуть включатися до складу інших (відношення типу "частина — ціле", "елемент — клас");
- 3) **зв'язність:** в інформаційній системі повинна бути передбачена можливість встановлення різних типів зв'язків між різними інформаційними одиницями (причинно-наслідкові, просторові та ін.);
- 4) **семантична метрика:** на множині інформаційних одиниць корисно задавати відношення, які характеризують ситуаційну близькість цих одиниць;
- 5) **активність:** виконання програм в інтелектуальній системі повинно ініціюватися поточним станом бази знань.

Часто виокремлюють інші властивості знань, наприклад **шкальованість**, яка означає, що формально неоднакові поняття насправді відображаються на одній і тій самій **шкалі понять**, різні точки якої відповідають інтенсивності того самого фактора. Так, температура може бути високою або низькою, і це породжує такі поняття, як "холодно", "тепло", "гаряче" тощо.

*Моделлю знань називається фіксована система формалізмів (понять і правил), відповідно до яких інтелектуальна система подає знання в своїй пам'яті та здійснює операції над ними.*

Моделі задання знань необхідні:

- для створення спеціальних мов описів знань і маніпулювання ними;
- для формалізації процедур зіставлення нових знань з уже існуючими;
- для формалізації механізмів логічного виведення.

Найвідомішими з моделей, які ґрунтуються на вербально-дедуктивній парадигмі, є чотири класи моделей:

- семантичні мережі;
- фреймові;
- логічні;
- продукційні.

Вербально-дедуктивні моделі задання знань мають багато спільних рис, Отже, можна вважати, що всі вони мають єдину концептуальну основу).

### 5.8. Неоднорідність знань. Області і рівні знань

Для успішного здійснення операцій зі знаннями необхідно відокремлювати в БЗ певні фрагменти, які називаються **областями знань**. Ці області знань повинні бути відносно незалежними між собою, що означає таке:

- зміни в одній області знань не повинні приводити до суттєвих змін у інших областях;
- вирішення складної задачі можна, як правило, звести до підзадач таким

чином, щодля вирішення кожної з цих підзадач достатньо знань з однієї області.

Такий розподіл є важливим для полегшення проектування і використання БЗ. Зокрема, різні області знань можуть проектуватися незалежно одна від одної.

Виходячи з постановки задачі можна по-різному розділяти знання на області. Так, для експертних систем, які ведуть діалог з користувачем мовою, наближеною до природної, можна виокремити такі області знань [7]:

- **предметна область**, яка містить знання про конкретну предмет, галузь, в якій працює експертна система;

- **область мови**, яка містить знання про мову, якою ведеться діалог;

- **область системи**, яка містить знання експертної системи про власні можливості;

- **область користувача**, яка містить знання про користувача. Наявність їх дозволяє враховувати індивідуальні особливості кожного користувача. Наприклад, пояснення користувачеві можуть надавати залежно від рівня його підготовленості;

- **область діалогу**, яка містить знання про мету діалогу, а також про форми та методи його організації.

Знання можуть різнитися за **рівнем задання і рівнем детальності**. За **рівнями задання** розрізняють знання нульового рівня (конкретні і абстрактні) і знання вищих рівнів — знання про знання (**метазнання**).

У випадку класифікації знань за **рівнями детальності** до уваги береться ступінь деталізації знань. Кількість рівнів детальності залежить від специфіки задачі, обсягу знань і моделі їх задання. Якщо користувач запитує в експертної системи, як здійснити певну операцію. У разі отримання відповіді на найвищому рівні детальності система видає *загальний план*. Якщо цей загальний план не є достатнім, система може спуститися на нижчий рівень і розписати операції детальніше.

### 5.9. База знань як об'єднання простіших одиниць

БЗ є кон'юнкцією простих тверджень. Можна вважати, що кожне з них відповідає окремому реченню природної мови. Подібні твердження називають **концептуальними одиницями**. Останні можуть бути або фактами, або правилами виведення.

Якщо концептуальна одиниця являє собою факт, вона може бути описана певним **предикатом**, тобто логічною функцією, що залежить від заданої кількості змінних і може приймати одне з двох можливих значень 0 або 1 (false або true). Конкретні твердження утворюються з предикатів шляхом підстановки конкретних значень аргументів. При цьому для опису одного й того самого твердження можна використовуватися різні предикати.

Так, два твердження:  $7 + 5 = 12$ ,

$8 + 9 = 15$

відповідають одному предикату Плюс( $a, b, c$ ).

Перша підстановка конкретних значень замість змінних  $a, b, c$  породжує істинне твердження, друга — хибне. Формально ці твердження будуть мати

запис: Плюс(7, 5, 12) та Плюс(8, 9, 15).

Можна також розглядати загальніший предикат: Операція (назва\_операції,  $a$ ,  $b$ ,  $c$ ), який можна використовувати для запису тверджень не тільки про додавання, але й про будь-яку іншу бінарну операцію: множення, ділення і т. п. Можливим є запис: Операція(Плюс, 7, 5, 12).

Який тип розглянутих предикатів слід обирати, залежить від проектувальника БЗ і специфіки конкретної задачі.

#### 5.10. Бінарні предикати і тріада "об'єкт—атрибут—значення"

Концептуальна одиниця може бути достатньо складною. Тому концептуальну одиницю, що є фактом, часто зручно розглядати як кон'юнкцію елементарніших тверджень, а саме — **бінарних фактів**, кожний з яких описується **бінарним предикатом**, тобто предикатом, який залежить від двох змінних. Якщо формалізувати правила об'єднання бінарних предикатів у складніші одиниці, на основі бінарних предикатів можуть бути сконструйовані складні твердження та системи знань.

Розглянемо приклад. Маємо твердження *"Студент Іванов отримав п'ятірку на іспиті зі штучного інтелекту"*.

Перепишемо цю фразу у вигляді кон'юнкції бінарних фактів:

Є(Іванов, Студент)Є(ІспШтІнт, Іспит)Ісп\_ШІ(Іванов, 5)

Предикат Ісп\_ШІ був введений для задання зв'язку між різними інформаційними одиницями.

У вигляді бінарного можна переписати й **унарний предикат**, тобто предикат, який залежить від однієї змінної. Наприклад, предикат Птах( $x$ ), який означає, що  $x$  є птахом, можна переписати як Є( $x$ , Птах).

Якщо певний унарний предикат описує **властивість** об'єкта, наприклад Літає( $X$ ), її можна переписати у вигляді Літає( $x$ , так).

Бінарним предикатам і бінарним фактам безпосередньо відповідає **тріада "об'єкт — атрибут — значення"**. Об'єкт у цій тріаді — це, як правило, назва деякої інформаційної одиниці, атрибут — назва певної ознаки, а значення — конкретне значення цієї ознаки для даного об'єкта. Тріада "об'єкт — атрибут — значення" є просто іншою формою заміни бінарного факту. Так, можна переписати наш приклад у вигляді:

Іванов — Є — Студент Ісп\_ШІ — Є — Іспит Іванов — Ісп\_ШІ — 5.

Остання тріада цього прикладу задає зв'язок між різними об'єктами.